

# On the Shoulders of Giants: Incremental Influence Maximization in Evolving Social Networks

Xiaodong Liu Xiangke Liao Shanshan Li Jingying Zhang Lisong Shao Chenlin Huang  
Liquan Xiao

National University of Defense Technology, Changsha, 410073, China

{liuxiaodong, xkliao, shanshanli, jyzhang, lisongshao, chenlinhuang, lqxiao}@nudt.edu.cn

## Abstract

Identifying the most influential individuals can provide invaluable help in developing and deploying effective viral marketing strategies. Previous studies mainly focus on designing efficient algorithms or heuristics to find top- $K$  influential nodes on a given static social network. While, as a matter of fact, real-world social networks keep evolving over time and a recalculation upon the changed network inevitably leads to a long running time, significantly affecting the efficiency. In this paper, we observe from real-world traces that the evolution of social network follows the preferential attachment rule and the influential nodes are mainly selected from high-degree nodes. Such observations shed light on the design of IncInf, an incremental approach that can efficiently locate the top- $K$  influential individuals in evolving social networks based on previous information instead of calculation from scratch. In particular, IncInf quantitatively analyzes the influence spread changes of nodes by localizing the impact of topology evolution to only local regions, and a pruning strategy is further proposed to effectively narrow the search space into nodes experiencing major increases or with high degrees. We carried out extensive experiments on real-world dynamic social networks including Facebook, NetHEPT, and Flickr. Experimental results demonstrate that, compared with the state-of-the-art static heuristic, IncInf achieves as much as  $21 \times$  speedup in execution time while maintaining matching performance in terms of influence spread.

**Categories and Subject Descriptors** G.2.2 [Graph Theory]: Graph algorithms and network problems

**General Terms** Algorithms, Performance

**Keywords** Influence maximization, incremental algorithm, evolving social network, graph algorithm

## 1. Introduction

Influence maximization (IM) is one fundamental and important problem which aims to identify a small set of influential individuals so as to develop effective viral marketing strategies in large-scale social networks[11]. As a matter of fact, real-world social networks keep evolving over time. For example, in Facebook, new people might join while old ones might withdraw, and people might make new friends with each other. Moreover, real-world social networks are evolving in a rather surprising speed; it is reported that as much as 1 million new accounts are created in Twitter every day [30]. Such massive evolution of network topology, on the contrary, may

lead to a significant transformation of the network structure, thus raising a natural need of efficient reidentification.

Existing researches and solutions on influence maximization focus mainly on developing effective and efficient algorithms on a given static social network. Although one could possibly run any of the static influence maximization methods, such as [7, 8, 24, 25], to find the new top- $K$  influential individuals when the network is updated, this approach has some inherent drawbacks that cannot be neglected: (1) the running time of a specific static method can be extremely long and unacceptable especially on large-scale social networks, and (2) whenever the network topology is changed, we need to recalculate the influence spreads for all the nodes which leads to very high costs. Can we quickly and efficiently identify the influential nodes in evolving social networks? Can we incrementally update the influential nodes based on previously known information instead of frequently recalculating from scratch?

Unfortunately, the rapidly and unpredictably changing topology of a dynamic social network poses several challenges in the reidentification of influential users, which we list as follows. On one hand, the interconnections between edges in real-world social graphs are rather complicated; as a result, even one small change in topology may affect the influence spreads of a large number of nodes, not to mention the massive changes in large-scale social networks. It is very difficult to efficiently compute the changes of influence spreads for all the nodes after the evolution. On the other hand, since there are a great number of nodes in large-scale social networks, how to effectively limit the range of potential influential nodes and reduce the amount of calculation to the maximum is a very challenging problem.

To well address these challenges, we investigate the dynamic characteristics exhibited during the evolution of real-world social networks. Through tests on three real-world dataset traces, Facebook, NetHEPT and Flickr, we observe that, first, the growth of social network is mainly based on the preferential attachment principle [3], that is the new-coming edges prefer to attach to nodes with higher degree, which naturally leads to the “rich-get-richer” phenomena; and second, the top- $K$  influential nodes are mainly selected from those high-degree nodes. Inspired by such observations, we know that the influence changes of some nodes will have no impact on the top- $K$  selection, and thus can be pruned to reduce the amount of calculation. Motivated by this, we propose IncInf, an incremental method to identify the top- $K$  influential nodes in evolving social networks instead of recalculating from scratch, thus significantly improving the efficiency and scalability to handle extraordinarily large-scale networks. To summarize, the main contributions of IncInf are as follows:

First, we design an efficient approach to quantitatively analyze the influence spread changes from network topology evolution by

adopting the idea of localization. A tunable parameter is provided to tradeoff between efficiency and effectiveness.

Second, we propose a pruning strategy which could effectively narrow the search space into nodes only experiencing major increases or with high degrees based on the changes of influence spread and the previous top- $K$  information.

Third, we conduct extensive experiments on three dynamic real-world social networks. Compared with the state-of-the-art static algorithm, IncInf achieves up to  $21\times$  speedup in execution time while providing matching influence spread. Moreover, IncInf provides better scalability to scale up to extraordinarily large-scale networks.

The remainder of this paper is organized as follows. Section 2 presents related preliminaries and problem definition. Section 3 shows the structural evolution characteristics of dynamic social networks that we observe from three datasets: Facebook, NetHEPT and Flickr. Section 4 details the design of our incremental algorithm IncInf. The performance of IncInf is evaluated by comprehensive experiments in Section 5. We present related work in Section 6 and conclude in Section 7.

## 2. Preliminaries and Problem Statement

In this section, we illustrate the definition of social network and the influence diffusion model that we will use throughout the paper, and then give the problem definition of influence maximization in evolving networks.

### 2.1 Preliminaries on Influence Maximization

**Social Network.** A social network is formally defined as a directed graph  $G = (V, E, P)$  where node set  $V = \{v_1, v_2, \dots, v_n\}$  denotes entities in the social network. Each node can be either active or inactive, and will switch from being inactive to being active if it is influenced by others nodes. Edge set  $E \subset V \times V$  is a set of directed edges representing the relationship between different users. Take Twitter as an example. A directed edge  $(v_i, v_j)$  will be established from node  $v_i$  to  $v_j$  if  $v_i$  is followed by  $v_j$ , which indicates that  $v_j$  may be influenced by  $v_i$ .  $P$  denotes the influence probability of edges; each edge  $(v_i, v_j) \in E$  is associated with an influence probability  $p(v_i, v_j)$  defined by function  $p : E \rightarrow [0, 1]$ . If  $(v_i, v_j) \notin E$ , then  $p(v_i, v_j) = 0$ .

---

#### Algorithm 1 Basic Greedy

---

```

1: Initialize  $S = \emptyset$ 
2: for  $i = 1$  to  $K$  do
3:   Select  $v = \arg \max_{v_i \in (V \setminus S)} (\sigma(S \cup v_i) - \sigma(S))$ 
4:    $S = S \cup \{v\}$ 
5: end for
```

---

**Independent Cascade (IC) Model.** IC model is a popular diffusion model that has been well-studied in [7, 17, 19, 25, 32]. Given an initial set  $S$ , the diffusion process of IC model works as follows. At step 0, only nodes in  $S$  are active, while other nodes stay in the inactive state. At step  $t$ , for each node  $v_i$  which has just switched from being inactive to being active, it has a single chance to activate each currently inactive neighbor  $v_j$ , and succeeds with a probability  $p(v_i, v_j)$ . If  $v_i$  succeeds,  $v_j$  will become active at step  $t+1$ . If  $v_j$  has multiple newly activated neighbors, their attempts in activating  $v_j$  are sequenced in an arbitrary order. Such a process runs until no more activations are possible [19]. We use  $\sigma(S)$  to denote the influence spread of the initial set  $S$ , which is defined as the expected number of active nodes at the end of influence propagation.

**Basic Greedy Algorithm.** Richardson and Domingos [11, 28] first introduced the influence maximization problem on static networks

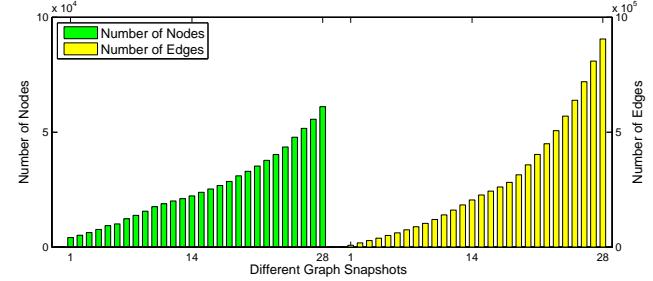


Figure 1: Number of Nodes and Edges per month of the Facebook dataset.

in 2001. In [19], Kempe et al. propose a basic hill-climbing greedy algorithm as shown in Algorithm 1. The proposed greedy algorithm works in  $K$  iterations, starting with an empty set  $S$  (line 1). In each iteration, a node  $v_i$  which brings the maximum marginal influence spread  $\sigma_S(v_i) = \sigma(S \cup v_i) - \sigma(S)$  is selected to be included in  $S$  (lines 3 and 4). The process ends when the size of  $S$  reaches  $K$  (line 2). However, this algorithm has a serious efficiency drawback due to the compute-intensive influence spread calculation. Several recent studies [7, 8, 14, 17, 18, 24, 25, 32] aimed at addressing this efficiency issue.

### 2.2 Formal Definition of IM problem in Evolving Networks

This paper differentiates itself from previous works by considering the dynamic nature of online social networks. As a matter of fact, the real-world social networks are not wholly static but keep evolving gradually over time. The evolution of large social networks has raised new sets of questions; among them one interesting yet challenging problem is how to quickly identify the top- $K$  influential users when the topology of the network is changed.

To solve such a problem, we define an evolving network  $\zeta = (G^0, G^1, \dots, G^t)$  as a sequence of network snapshots evolving over time, where  $G^t = (V^t, E^t, P^t)$  is the network snapshot at time  $t$ .  $\Delta G^t = (\Delta V^t, \Delta E^t, \Delta P^t)$  denotes the structural change of network graph  $G^t$ . Obviously, we have  $G^{t+1} = G^t \cup \Delta G^t$ . And the influence maximization problem is defined as follows:

**Given:** The social network  $G^t$  at time  $t$ , the top- $K$  influential nodes  $S^t$  in  $G^t$ , and the structural evolution  $\Delta G^t$  of graph  $G^t$ .

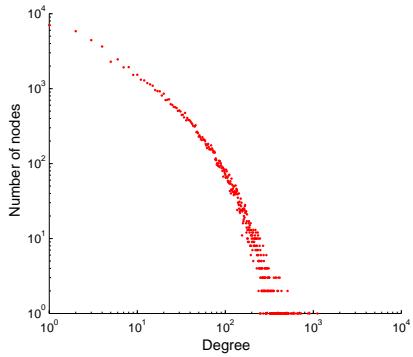
**Objective:** To identify the influential nodes  $S^{t+1} \subset V^{t+1}$  of size  $K$  in  $G^{t+1}$  at time  $t+1$ , such that the influence spread  $\sigma(S^{t+1})$  is maximized at the end of influence diffusion.

## 3. Observations of Social Network Evolution

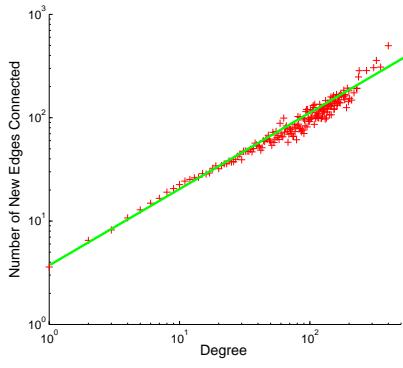
In this section, we study some patterns of social network evolution. The number of nodes and edges are firstly investigated in Section 3.1 to examine the growth of users and interconnections over time. Then, we look into the degree distribution of nodes and the preferential attachment rule for new edges in Section 3.2. We further examine the relation between the influence and the degree of node in Section 3.3. We study three network traces: Facebook, NetHEPT and Flickr whose detailed description can be found in Section 5. Here we only show the results on Facebook since the evolution trends on the other datasets are qualitatively similar and thus omitted.

### 3.1 How Fast does the Network Evolve?

Nodes and edges are the basic elements of the social network topology. In this subsection, we use the number of nodes and edges to examine the growth of users and interconnections over time.



(a) Degree distribution



(b) Preferential attachment

Figure 2: Degree distribution and preferential attachment on Facebook.

Figure 1 illustrates the number of nodes and edges over the entire trace period on the Facebook dataset; we take a snapshot per month. From Figure 1, we observe a linear increase in the number of nodes which indicates a steady number of new users joined the network per month. While in terms of edges, the number goes up almost exponentially. The number of edges after 14 months is  $25.6 \times$  of that in the initial graph while the number rises to  $112.9 \times$  after 28 months. Such rapid growth of nodes and edges raise a natural need to efficiently find the most influential nodes after the topology evolution.

### 3.2 What is the Pattern of Network Topology Evolution?

Understanding the pattern of the network topology evolution is of primary importance to design efficient influence maximization algorithms for evolving social networks. In this subsection, we further investigate the degree distribution of nodes and the preferential attachment rule [3, 4, 23] for new coming edges. Figure 2a shows the degree distribution of the Facebook final graph in log-log scale. As expected, it mainly follows the well-known power-law distribution. A large percent of the users have only a small number of links with other users, while there exist some “hub” nodes with extremely large number of connections. This is consistent with the real-world networks.

We also study the preferential attachment rule, or in other words, the “rich-get-richer” rule [12, 20], which postulates that when a new node joins the network, it creates a number of edges, where the destination node of each edge is chosen proportional to the destination’s degree. This means that new edges are more likely

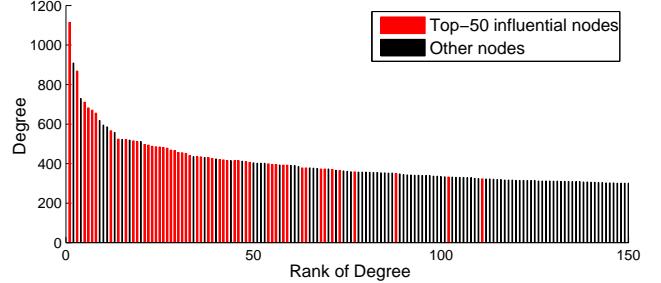


Figure 3: The relation between the influence spread and the degree in Facebook.

to connect to nodes with high degree than ones with low degree. This is reasonable in reality; Lady Gaga gains 30,000 new followers on average every day [21] which can never image for any common individual. The results on the Facebook dataset are demonstrated in Figure 2b where the x axis is the degree of different nodes and the y axis is the average number of new edges attached to nodes of different degree. Note that both the x and y axis are in log scale. From Figure 2b we can see that the degree of users in Facebook is linearly correlated with the number of new links created. This suggests that high-degree nodes get super-preferential treatment. Consequently, the influence spread change should be considerably great for the influential nodes, while there may be only small or even no change for ordinary people.

### 3.3 What is the Relation between Influence and Degree?

Examining the relation between the influence and the degree of node can help us understand the effect of degree changing on the influence spread of nodes. For this reason, we run the static MixGreedy algorithm [7] on the final graph and identify the top-50 influential nodes. The results on the Facebook dataset are illustrated in Figure 3 where the x axis is the rank of degrees of different nodes (we only show the top 150). Obviously, all the selected influential nodes have a large degree. In particular, among the 50 nodes, 48 nodes rank in top 100 of the whole 61,096 nodes in terms of degree, and the other two nodes rank 102 and 111 respectively. While on the NetHEPT and Flickr datasets, the top-50 influential nodes are selected from the top 1.79% and 0.84% nodes in degree, respectively. This demonstrates that the top- $K$  influential nodes are mainly selected from those with large degrees. However, it is worthy of note that the top- $K$  influential nodes in influence maximization are usually not the top- $K$  nodes ranking in degree, since the influence spread of different nodes may overlap with each other.

## 4. IncInf Design

In this section, we present the detailed design of IncInf, an incremental approach to solve the influence maximization problem on dynamic social networks. The main idea of IncInf is to take full use of the valuable information that is inherent in the network structural evolution and previous influential nodes, so as to substantially narrow the search space of influential nodes. In this way IncInf can significantly reduce the computation complexity and improve the efficiency. Figure 4 briefly illustrates the general idea of IncInf in dynamic social networks. The top- $K$  influential nodes  $S^{t+1}$  of  $G^{t+1}$  at time  $t+1$  is incrementally identified based on the previous influential nodes  $S^t$  at time  $t$  and the structural change  $\Delta G^t$  from  $G^t$  to  $G^{t+1}$ . In particular, we design an efficient method to quantitatively analyze the impact of different structural changes on the

Table 1: Details of six types of basic operation

Operation	Description	Impact on influence spread
$\text{addNode}(u)$	add a new node $u$ into the current network	the influence spread of $u$ is set to 1
$\text{removeNode}(u)$	delete an existing node $u$ from the network	the influence spread of $u$ is set to 0
$\text{addEdge}(u, v, w)$	introduce a new edge $(u, v)$ with $p(u, v) = w$	the influence spread of all the nodes that can reach $u$ may be increased
$\text{removeEdge}(u, v)$	remove an existing edge $(u, v)$ from the network	the influence spread of all the nodes that can reach $u$ may be decreased
$\text{addWeight}(u, v, \Delta w)$	increase $p(u, v)$ by $\Delta w$	the influence spread of all the nodes that can reach $u$ may be increased
$\text{decWeight}(u, v, \Delta w)$	reduce $p(u, v)$ by $\Delta w$	the influence spread of all the nodes that can reach $u$ may be decreased

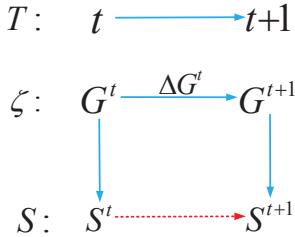


Figure 4: InCInf Design.

influence spread of nodes by adopting the idea of localization (Section 4.2), and propose a pruning strategy to reduce the number of potential influential nodes (Section 4.3). We first describe six types of basic operation of topology evolution in dynamic networks in Section 4.1.

#### 4.1 Basic operations of Topology Evolution

The evolution of social network, when reflected into its underlying graph, can be summarized into six categories, which are inserting or removing a node, introducing or deleting an edge, and increasing or decreasing the influence probability of an edge. We denote the six types of topology change as  $\text{addNode}$ ,  $\text{removeNode}$ ,  $\text{addEdge}$ ,  $\text{removeEdge}$ ,  $\text{addWeight}$ ,  $\text{decWeight}$ . The detailed descriptions and their effects on influence spread are shown in Table 1.

It should be noted that only after the  $\text{addNode}$  operation can node  $u$  establish links ( $\text{addEdge}$ ) or sever links ( $\text{removeEdge}$ ) with other nodes, and node  $u$  can only be removed when all its associated edges are deleted. Moreover, the weight operation can be equivalently decomposed into two edge operations. For example,  $\text{addWeight}(u, v, \Delta w)$  can be divided into  $\text{removeEdge}(u, v)$  and  $\text{addEdge}(u, v, w + \Delta w)$ , supposing the previous weight of edge  $(u, v)$  is  $w$ .

#### 4.2 Influence Spread Changes

As discussed above, whenever an edge  $(u, v)$  is introduced into or removed from the social network, the influence spread of all the nodes that can reach node  $u$  may be changed. However, as a matter of fact, the real-world social networks exhibit small-world network characteristics and the connections between nodes are highly complicated. As a result, even one small change in topology, such as an edge addition or removal, may affect the influence spread of a large number of nodes, thus introducing massive recalculations. In order to reduce the amount of computation, we design an approach to efficiently calculate the changes on the influence spread of nodes

which adopts the localization idea [8] and tries to restrict the influence spread to the local regions of nodes.

The main idea of localization is to use the local region of each node to approximate its overall influence spread. In particular, we use the maximum influence path to approximate the influence spread from node  $u$  to  $v$ . Here the maximum influence path  $MIP(u, v, G)$  from node  $u$  to  $v$  in graph  $G$  is defined as the path with the maximum influence probability among all the paths from node  $u$  to  $v$ , and can be formally described as follows:

$$MIP(u, v, G) = \arg \max_{p \in P(u, v, G)} \{prob(p)\} \quad (1)$$

where  $prob(p)$  denotes the propagation probability of path  $p$  and  $P(u, v, G)$  denotes all the paths from node  $u$  to  $v$  in graph  $G$ . For a given path  $p = \{u_1, u_2, \dots, u_m\}$ , the propagation probability of path  $p$  is defined as follows:

$$prob(p) = \prod_{i=1}^{m-1} p(u_i, u_{i+1}) \quad (2)$$

Moreover, an influence threshold  $\theta$  is set to tradeoff between accuracy and efficiency. During the propagation process, we only consider paths whose influence probability are larger than  $\theta$  while ignoring those with probability smaller than  $\theta$ . By doing this, the influence is effectively restricted to the local region of each node.

Similarly, in our proposal we localize the impact of topology changes on influence spread into local regions, and thus reduce the amount of computation. Among six types of topology change,  $\text{addNode}$  (or  $\text{removeNode}$ ) is the most straightforward since it simply sets the influence spread of the node to 1 (or 0);  $\text{addWeight}$ ,  $\text{decWeight}$  as well as  $\text{removeEdge}$  are methodologically similar to  $\text{addEdge}$ . Consequently, in the following we take  $\text{addEdge}$  as an example to show which nodes' influence spread need to be updated and how to determine those changes when a new edge is added into the graph.

Consider the case when a new edge  $e = (u, v, w)$  is introduced between two existing node  $u$  and  $v$ . We denote the graph before and after such a topology change as  $G^t$  and  $G^{t'}$ , and the current seed set is  $S$ . The detailed algorithm is described in algorithm 2. According to the principle of localization [8], if the propagation probability  $w$  is smaller than the specified threshold  $\theta$ , or not bigger than the probability of  $MIP(u, v, G^t)$ , edge  $e$  can be simply neglected and there is no need to update any node's influence spread (lines 1-3). Otherwise, the newly-added edge  $e$  would become the  $MIP(u, v, G^{t'})$ . As a result, each node  $i$  whose maximum influence path to  $u$  has a influence probability larger than  $\theta$  is likely to experience a rise in terms of influence spread (line 4) because node  $i$  may influence more nodes through the new edge  $e$ . So, we then check the probability of the maximum influence path from  $i$  to

---

**Algorithm 2** Edge addition

---

**Input:** a new edge  $e = (u, v, w)$ , graph  $G^t$ .  
**Output:** The influence spread changes of nodes in  $G^{t'}$ .

```

1: if  $w < \theta$  or  $w \leq \text{prob}(\text{MIP}(u, v, G^t))$  then
2:   return;
3: end if
4: for each node  $i$  with  $\text{prob}(\text{MIP}(i, u, G^t)) > \theta$  do
5:   for each node  $j$  with  $\text{prob}(\text{MIP}(v, j, G^t)) > \theta$  do
6:     if  $\text{prob}(\text{MIP}(i, j, G^t)) < \theta$  and
       $\text{prob}(\text{MIP}(i, j, G^{t'})) > \theta$  then
7:        $\text{deltaInf}[i] += \text{prob}(\text{MIP}(i, j, G^{t'})) \times (1 - \text{prob}(j, S))$ 
8:     end if
9:     if  $\text{prob}(\text{MIP}(i, j, G^t)) > \theta$  and
       $\text{prob}(\text{MIP}(i, j, G^{t'})) > \theta$  then
10:       $\text{deltaInf}[i] += (\text{prob}(\text{MIP}(i, j, G^{t'})) - \text{prob}(\text{MIP}(i, j, G^t))) \times (1 - \text{prob}(j, S))$ 
11:    end if
12:  end for
13: end for

```

---

$v$  and its successors in  $G^t$  and  $G^{t'}$ . Based on the two probabilities, we divide the problem into two small cases:

The first case is when the probability of maximum influence path from  $i$  to  $j$  in  $G^t$  is smaller than  $\theta$  while that in  $G^{t'}$  is larger than  $\theta$  (lines 5-6). Here  $j$  denotes the node whose probability of  $\text{MIP}(v, j, G^t)$  is larger than  $\theta$ . In such a case, node  $i$  build a new path to  $j$  through the new edge  $e$  which increases the influence spread of  $i$  by  $\text{prob}(\text{MIP}(i, j, G^{t'})) \times (1 - \text{prob}(j, S))$  (line 7). Here  $\text{prob}(j, S)$  is the probability of that node  $j$  is influenced by the current seed set  $S$ , which is defined as follows:

$$\text{prob}(j, S) = \begin{cases} 1, & \text{if } j \in S \\ 1 - \prod_{w \in n(j)} 1 - \text{prob}(w, S) \cdot p(w, j), & \text{if } j \notin S \end{cases}$$

Here  $n(j)$  denotes the in-neighbour set of  $j$ .

The second case is when the probability of maximum influence path from  $i$  to  $j$  is larger than  $\theta$  in both  $G^t$  and  $G^{t'}$  (lines 9-11). In this case, the influence increase of node  $i$  is  $(\text{prob}(\text{MIP}(i, j, G^{t'})) - \text{prob}(\text{MIP}(i, j, G^t))) \times (1 - \text{prob}(j, S))$ .

We treat the network dynamics from  $G^t$  to  $G^{t+1}$  as a finite change stream  $c_1, c_2, \dots, c_i, \dots$  where each change  $c_i$  is one of the six topology changes we described above. When all the changes in the change stream are processed, we can obtain the influence spread change for all the nodes.

### 4.3 Potential Top- $K$ Influential Users Identification

Inspired by the observations of Section 3, we design a pruning strategy to reduce the search space of potential influential nodes in this subsection. It is assumed that we only know who are the top- $K$  influential nodes in graph  $G^t$ , but their detailed influence spreads are beyond our knowledge. The reason are mainly twofold. First, several influence maximization algorithms, such as DegreeDiscount [7] and SA [17], do not calculate the influence spread information to identify influential users so that such information are unavailable. Second, even though these information are ready, storing them will cost as much as  $O(nK)$  memory space where  $n$  is the number of node in  $G^t$ . Since real-world social networks are typically of large scale, this will introduce serious storage overhead and directly affect the scalability.

From the preferential attachment rule, we know that the influence spread changes of those high-degree nodes should be much

greater than the ordinary nodes. Moreover, according to the power-law distribution, such high-degree nodes only account for a small part of the whole nodes. Consequently we can pick out nodes only experiencing major increases or with high degrees because these nodes are of great potential to become the top- $K$  influential nodes in  $G^{t+1}$ . Then we only calculate the actual influence spread for these selected nodes while ignoring the others. In this way, a large percent of nodes are pruned and the search space is largely narrow. It should be noted that a smart pruning strategy is of key importance since a poor selection might either affect the efficiency or reduce the accuracy in terms of influence spread. We describe the details of our pruning strategy as follows:

1. In the  $i$ th iteration, if the influence spread of the previous influential node  $S_i^t$  increases in  $G^{t+1}$ , the chosen nodes are those with a larger influence spread change than  $\text{deltaInf}[S_i^t]$ ;

In most cases, the influential nodes will attract a great number of new nodes and establish new links. Thus, their influence spreads will increase drastically. In such a case, the nodes whose influence spread changes are smaller than the influential nodes are completely impossible to become the most influential node in  $G^{t+1}$ . Therefore, when the influence spread of the previous influential nodes increase, we only select those whose influence spread changes are larger than the influential nodes in  $G^t$ . According to the preferential attachment rule, such a pruning method can greatly narrow the search space and reduce the amount of computation.

2. In the  $i$ th iteration, if the influence spread of the previous influential node  $S_i^t$  decreases in  $G^{t+1}$ , in addition to qualification 1, the nodes are further selected to hold a sufficiently large degree or experience a sufficiently great increase. In order to formally define “large degree” and “great increase”, here we set an threshold  $\eta$  to tradeoff between running time and influence spread. Here the nodes with sufficiently large degrees (or great increase) are defined as the set of node  $v_j$  whose degree (or degree increase ratio) is among the top  $\eta$  percent of all nodes in  $G^{t+1}$ . The degree increase ration of  $v_j$  is defined as  $\text{degree}_j^{t+1}/\text{degree}_j^t$  where  $\text{degree}_j^t$  denotes the degree of node  $v_j$  in graph  $G^t$ . Experimental results in Section 5 will demonstrate that 5% may stand as a good tradeoff between running time and influence spread.

It should be noted that although the case the influence spread of a previous influential node decreases during the evolution rarely happens, we consider it here for completeness. In this case, except for qualification 1, we further select nodes because the number of nodes satisfying qualification 1 is relatively large which lead to mass computation. While in reality, a node with small degree has only very low probability to become an influential node. In order to select only the most potential nodes, we refine the requirement and additionally select the nodes with large degree and large increase. Consequently, the search space is strictly circumscribed and the computational complexity is greatly reduced.

After the potential nodes are selected, we calculate the actual influence spread of these nodes in  $G^{t+1}$  and select the one with the maximum influence spread in each iteration. Algorithm 3 outlines the design of our proposed algorithm IncInf. IncInf iterates for  $K$  round (line 2) and in each round select one node providing the maximum marginal influence spread. Lines 3 - 5 calculate the influence spread change of each node caused by the topology evolution. Nodes with great potential to become top- $K$  influential are selected (line 6) and their influence spread are computed in  $G^{t+1}$  (lines 7 - 9). And then the node providing the maximal

Table 2: Summary information of the real-world social networks

Datasets	Nodes			Edges		
	Initial Number	Final Number	Growth	Initial Number	Final Number	Growth
Facebook	12,364	61,096	394%	73,912	905,665	1125%
NetHEPT	5,802	29,555	409%	57,765	352,807	511%
Flickr	1,620,392	2,570,535	58.6%	17,034,807	33,140,018	94.5%

---

### Algorithm 3 InclInf

---

**Input:**  $G^t$ ,  $S^t$ , and  $G^{t+1}$ .  
**Output:** the top- $K$  influential nodes  $S^{t+1}$  in  $G^{t+1}$ .

```

1: Initialize  $S^{t+1} = \emptyset$ ;
2: for  $i = 1$  to  $K$  do
3:   for each topology change  $c_j$  from  $G^t$  to  $G^{t+1}$  do
4:     calculate the influence spread change  $\delta_{\text{Infl}}[\cdot]$ ;
5:   end for
6:   select a set of potential nodes as  $pn$  according to pruning
strategy;
7:   for each node  $v_i \in pn$  do
8:     calculate the marginal influence spread  $\sigma_{S^{t+1}}(v_i)$ ;
9:   end for
10:  select  $v_{max} = \arg \max_{v_j \in pn} (\sigma_{S^{t+1}}(v_j))$ ;
11:   $S = S \cup v_{max}$ ;
12: end for

```

---

marginal gain will be selected and added to the set  $S^{t+1}$  (lines 10 - 11).

## 5. Experiments

In this section, we present the experimental results of our algorithm on identifying top- $K$  influential nodes in dynamic social networks. We examine two metrics, running time and influence spread, for evaluating the effectiveness as well as the execution efficiency of different algorithms. The experimental results are detailed in Section 5.2, 5.3 and 5.4.

### 5.1 Experimental Setup

We choose three real-world social networks including Facebook social network, NetHEPT citation network, and Flickr social network. Table 2 summarizes the statistical information of the datasets.

- **Facebook.** This dataset is the friendship relationship network among New Orleans regional network on Facebook, spanning from Sep 2006 to Jan 2009 [31]. There are more than 60K users connected together by as much as 1.5M links in the social network. 41.4% of these edges contain no time information and are thus discarded. In our experiments, the nodes and links from Sep. 2006 to Apr. 2007 are used as the first snapshot and then network snapshots are recorded every 3 months.
- **NetHEPT.** This is an academic citation network [2] extracted from “High Energy Physics-Theory” section of the arXiv over the period from 1992 to 2003, and covers the citations within a dataset of 28K papers with 352K edges. In our experiments, the citation links of the first three year (i.e. from 1992 to 1994) are considered as the basic graph and the network snapshots are recorded once a year.
- **Flickr.** This dataset [27] contains the user-to-user links crawled from the Flickr social network daily over the period from Nov. 2, 2006 to Dec. 3, 2006 and again from Feb. 3, 2007 and May 18, 2007, representing a total of 104 days of growth. There are totally 2.5M Flickr users and 33M links. During this period

of observation, over 9.7 million new links are formed and over 950,000 new users joined the network. In our experiments, we use the network before Nov. 2, 2006 as the basic graph and another five snapshots are recorded on Dec. 3, Feb. 3, Mar. 3, Apr. 3, and May 18.

We compare our algorithm with four static algorithms: **Mix-Greedy**, **ESMCE**, **MIA** and **Random**. MixGreedy is an improved greedy algorithm on the IC model proposed by Chen et al. in [7]. ESMCE is a power-law exponent supervised estimation approach designed by Liu et al. in [25]. MIA is a heuristic that uses local arborescence structures of each node to approximate the influence propagation [8]. Random is a basic heuristic that randomly selects  $K$  nodes from the whole datasets.

The propagation probability of the IC model is selected randomly from 0.1, 0.01, and 0.001 for each network snapshot, and we run simulations on networks 10000 times and take the average of the influence spread.

### 5.2 Efficiency Study

In this subsection, the efficiency of our proposed algorithm is studied and compared with corresponding static algorithms, Mix-Greedy and MIA, through experiments on the Facebook, NetHEPT and Flickr datasets. The experiments are conducted on a PC with Intel Core i7 920 CPU @2.67 GHz and 6 GB RAM. The running time of four algorithms are measured by selecting 50 seeds from the whole dataset.

The time costs of different algorithms are illustrated in Figure 5 where we record the total time cost for each snapshot of the three datasets. Since incremental and static algorithms have the same time cost in the initial snapshot, thus they are omitted in the figure. The experimental results show that the time costs of our algorithm on each snapshot are obviously less than those of static algorithms. Obviously, MixGreedy takes the longest time among four kinds of influence maximization algorithms. It takes MixGreedy more than as much as 6 hours to identify the top 50 influential nodes on the final NetHEPT dataset, while the time is even longer on the larger dataset Facebook. Moreover, MixGreedy is not feasible to run on the largest dataset Flickr due to the unbearably long running time. ESMCE, benefiting from its sampling estimation method, runs much faster than MixGreedy, but it still takes as much as 3511 seconds on average to run on the five snapshots of Flickr. Compared with two greedy algorithms, the heuristic MIA performs much better. It only takes MIA 23.8 seconds to run on the final Facebook graph. When running on the Flickr dataset with as much as 2.5M nodes and 33M edges, however, its speedup is far from satisfactory, since it still needs more than 45 minutes to finish. While our proposed algorithm, IncInf, outperforms all the static algorithms in terms of efficiency. In particular, IncInf is almost four orders of magnitude faster than the MixGreedy algorithm on the Facebook dataset. While compared with the MIA heuristic, the speedup of IncInf is 8.41× and 6.94× on the Facebook and NetHEPT datasets, respectively; What’s more, when applied on the largest dataset Flickr, IncInf can achieve as much as 20.65× speedup on average. This is because IncInf only computes the incremental influence spread changes and adaptively identifies the

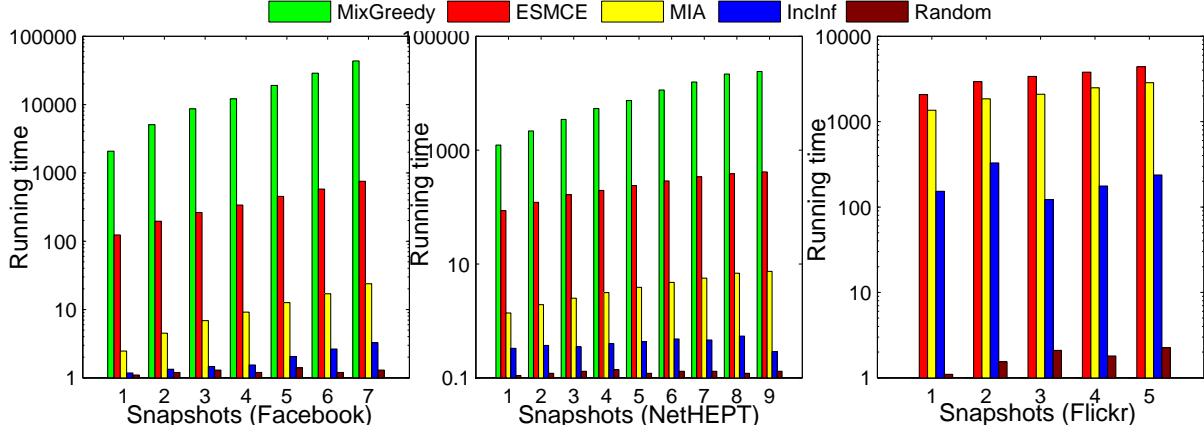


Figure 5: The time costs of different algorithms on three real-world datasets.

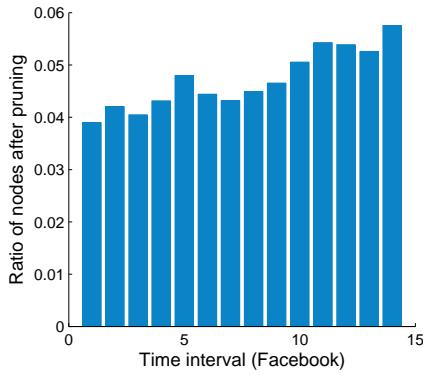


Figure 6: The effect of pruning strategy on the Facebook dataset.

influential nodes based on the previous influential nodes and the current influence spread changes. The experimental results clearly validate the efficiency advantage of our incremental algorithm IncInf. We can also observe that the running time of IncInf is not monotone like other algorithms as the time evolves. This is because the running time of IncInf is closely related to the topology change between two graph snapshots. An evident change in topology will usually lead to a relatively long running time and vice versa. Without doubt, Random runs the fast among all the algorithms. However, as we will show in Section 5.3, its accuracy is much worse and unacceptable when developing real-world viral marketing strategies.

We also test the effect of our pruning strategy. Here we take the Facebook dataset as an example; the results on other datasets are similar and thus omitted. Different from other experiments, we recorded the Facebook graph from Sep. 2006 to Oct. 2007 (14 months) as snapshot  $A$  in this experiment. After that we take snapshots every month as snapshot  $B$ . We use IncInf to find the top- $K$  influential nodes in snapshot  $B$  based on ones in snapshot  $A$ . The result is shown in Figure 6. The x axis is the time interval between snapshot  $A$  and  $B$ , and the y axis is the ratio of the number of nodes after pruning to the total number of nodes in snapshot  $B$ . The minimum and maximum pruning ratios are 3.90% and 5.86% respectively, with a mean ratio of 4.72% on all the 14 time intervals between snapshot  $A$  and  $B$ . This demonstrates that our pruning strategy can effectively limit the search space into a small percent of nodes. We can also see in Figure 6 that with the increase of

time interval, the ratio, although not monotone, generally becomes larger. This is mainly because a longer time interval means a larger amount of topology changes, and basically more nodes will be potential to become influential nodes.

### 5.3 Effectiveness Study

In this subsection, we study the influence spread of the top- $K$  influential nodes selected by our algorithm as well as other static algorithms. The influence spreads of different algorithms are measured as the number of nodes that are influenced by the top-50 influential nodes selected. Obviously, the higher the influence spread, the better the effectiveness. We have not tested the performance of MixGreedy on the Flickr dataset as the running time is excessively long.

Figure 7 shows the experimental results. MixGreedy outperforms all the other algorithms in terms of influence spread. However, the efficiency issue limits its application to large-scale dataset such as Flickr. The performance of ESMCE, MIA and IncInf almost match MixGreedy on the Facebook dataset, while on NetHEPT, the gaps become larger but remain acceptable (only 3.4%, 4.7% and 5.1% lower than MixGreedy on average). When applied to the Flickr dataset, ESMCE performs the best since ESMCE strictly control the error threshold by iterative sampling. Compared with MIA, IncInf shows very close performance and is only 2.87% lower on average of all five snapshots, which demonstrates the effectiveness of our proposal. Random, as the baseline heuristic, clearly performs the worst on all the graphs. Actually, the influence spread of Random is only 15.6%, 12.1% and 10.9% of that of IncInf on Facebook, NetHEPT and Flickr, respectively.

We shall note that the reason IncInf has slightly lower influence spread is mainly twofold. First, IncInf restrict the influence into local regions to speed up the computation of influence spread changes, which will affect the effectiveness. Second, a pruning strategy is designed to narrow down the search space based on the influence spread changes and previous top- $K$  information. Despite slight loss in effectiveness, as aforementioned, the disparity is small and acceptable. More importantly, IncInf gains remarkable improvement in efficiency.

### 5.4 Tuning of Parameter $\theta$ and $\eta$

First, we study how effectively the localization parameter  $\theta$  of IncInf represents a tradeoff between efficiency and effectiveness. We run IncInf with different values of  $\theta$  on the final Facebook and NetHEPT graphs. The running time and influence spread are measured based on seed size  $K = 50$ .

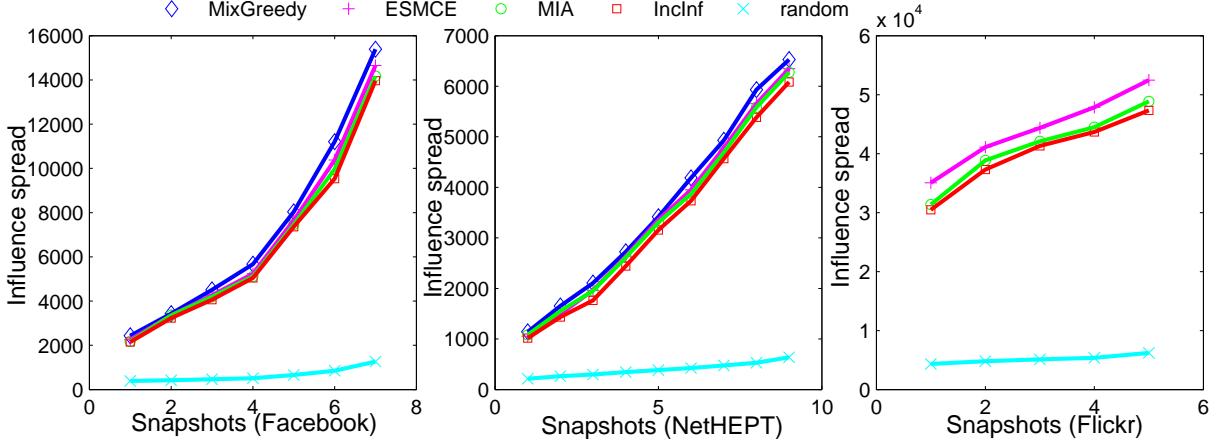


Figure 7: The influence spread of different algorithms on three datasets.

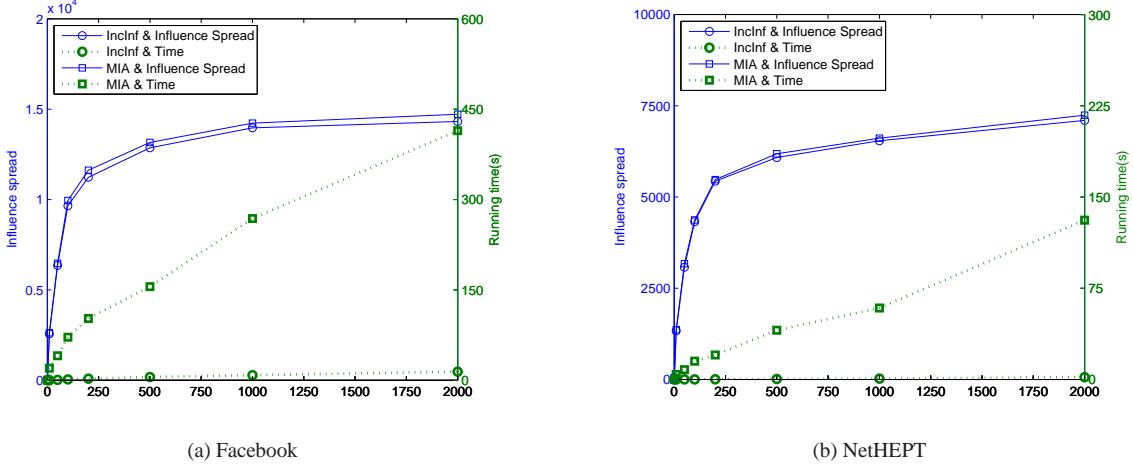


Figure 8: The effect of tuning of  $\theta$  on running time and influence spread.

The experimental results are shown in Figure 8. Note that the x axis represents the reciprocal of  $\theta$ . We observe that  $\theta$  acts as a tradeoff between efficiency and effectiveness: with the decrease of  $\theta$ , IncInf and MIA achieve better influence spread. However, this is gain at the cost of longer running time, i.e., poor efficiency. For example, when we reduce  $\theta$  from 1/200 to 1/500 on the Facebook dataset, the influence spread of IncInf increases by 15.4% while the running time is 1.12 $\times$  longer. Moreover, we can observe that the influence spread of IncInf almost match that of MIA in all values of  $\theta$ . For example, IncInf is only 1.87% lower than MIA in influence spread when  $\theta$  is set to 1/200 in the NetHEPT dataset. But IncInf shows overwhelming advantages in terms of running time. When  $\theta$  is set to 1/500 in Facebook, IncInf needs only 5 second to identify the top-50 influential nodes while it takes MIA more than 150 second to finish the same work. More importantly with the decrease of  $\theta$ , the influence spread increases sharply at the beginning but the increase is no longer that significant after  $\theta$  is lowered to a certain level. On the contrary, the running time is almost linear to  $1/\theta$ . This suggests that the knee point of the influence spread curve can serve as a good tuning point of  $\theta$  where we could obtain the best gain from both influence spread and running time.

Then, we will evaluate the sensitivity of pruning threshold  $\eta$  in terms of influence spread and running time. The results are illustrated in Figure 9. From figure 9 we can see that, with the increase of  $\eta$ , the running time increase gently at the beginning and then turns into a sharp boost. For example, when we increase  $\eta$  from 1% to 5%, the running time of IncInf on the Facebook dataset only increase from 2.13s to 8.47s, while it dramatically increases from 8.47s to 87.35 when  $\eta$  is tuned from 5% to 10%. This phenomenon is closely related to the power-law distribution of degree in social network; when  $\eta$  set large, a relatively large number of potential nodes would be selected.

In terms of influence spread, as the increase of  $\eta$ , more nodes are selected as potential nodes which will guarantee better influence spread. Different from the running time, the influence spread grows rather rapid at the beginning, and then gradually slows down. The influence spread on the Facebook Dataset is 7854 when  $\eta$  is set to 1%, and rapidly grow to 13967 when the maximum error threshold is 5%. After that, the growth trend slows down and the influence spread is about 15091 as  $\eta$  increases to 10%. This reason to explain such phenomenon is that the top- $K$  influential nodes are mainly selected from high degree nodes. Therefore, when  $\eta$  becomes larger,

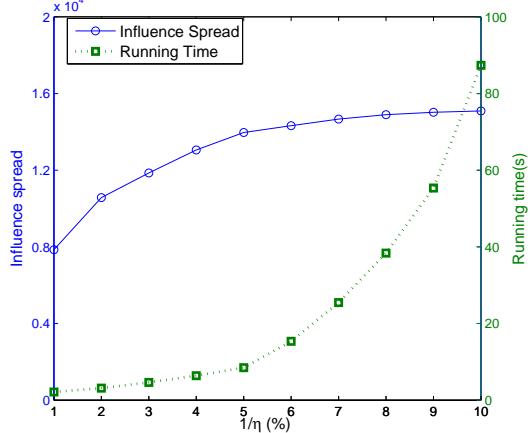


Figure 9: The effect of tuning of  $\eta$  on running time and influence spread.

although more nodes would be selected, their contribution to influence spread are relatively small, thus the growth trend slows down. Based on the above observation, here we suggest that 5% may stand as a good tradeoff between running time and influence spread.

## 5.5 Discussions

Experimental results demonstrate that our proposed IncInf algorithm significantly reduces the execution time of state-of-the-art static influence maximization algorithm while maintaining satisfying accuracy in terms of influence spread. Although IncInf performs better, it has a few limitations for further improvement.

First, IncInf directly depends on previous information of top- $K$  influential nodes for effective pruning, while sometimes such information are incomplete, or even unavailable. We plan to study this problem later. Second, IncInf is designed for the IC model which may somehow limit its application. But we believe our idea of incremental computation for influence maximization could be properly extended to other influence diffusion models.

## 6. Related Work

Influence maximization on static networks has attracted a lot of attentions. The hill-climbing greedy algorithms proposed by Chen et al. suffers from low efficiency, and many efficient algorithms have been proposed recently to address this problem. Leskovec et al. [24] exploit the submodularity of influence spread function and develop an optimized greedy algorithm, CELF, which is much faster than basic greedy algorithm. Chen et al. [7] propose MixGreedy which computes the influence spread for each seed set in one single simulation and incorporates the CELF optimization. MIA [8] uses local arborescence structures of each node to approximate the influence spread, thereby gaining efficiency by restricting computations and updates only on the local regions. However, MIA only considers static networks while in this paper we specifically design an incremental algorithm for evolving social networks. Recently, Wang et al. [32] propose a Community Greedy Algorithm (CGA) that took community property into account. Goyal et al. propose CELF++[15] further exploits the property of submodularity of the spread function to avoid unnecessary re-computations of marginal gains, and considerably improves the efficiency of CELF algorithm. IRIE [18] is also a heuristic proposed by Jung et al. that incorporates influence ranking algorithm with influence estimation method to achieve scalability. Chen et al. [9] propose a BatchGreedy algorithm for active learning and demonstrated through experiments

that BatchGreedy could considerably improved the effectiveness of previous greedy algorithms. Liu et al. [26] design a new framework to accelerate the influence maximization by leveraging the parallel processing capability of GPU. In [22], Lee et al. propose GIS with a similar idea of influence localization, but they didn't consider the dynamic feature of online social networks. Cheng et al. [10] present IMRank to solve the IM problem via finding a self-consistent ranking.

The influence maximization problem on dynamic social networks still remains largely unexplored to date. Habiba et al. [16] propose a dynamic social network model which is different from ours. In their proposal, the network keeps evolving during the process of influence propagation, and their goal is to find the top- $K$  influential nodes over such a dynamic network. When compared to [16], our work is based on snapshot graph model and our goal is to incrementally identify top- $K$  influential nodes based on the topology changes of two adjacent snapshots. Chen et al. [6] extend the IC model to incorporate the time delay aspect of influence diffusion among individuals in social networks, and consider time-critical influence maximization, in which one wants to maximize influence spread within a given deadline. While in [13], the authors consider a continuous time formulation of the influence maximization problem in which information or influence can spread at different rates across different edges. Charu Aggarwal et al. [1] try to discover influential nodes in dynamic social networks and they design a stochastic approach to determine the information flow authorities with the use of a globally forward approach and a locally backward approach. Their influence model and target are different from ours. Zhuang et al. [33] argue that the evolution of online social network could not be fully observed and focus on the problem of designing a proper probing strategy so that the actual influence diffusion process can be best uncovered with the probing nodes.

## 7. Conclusion and Future Work

In this paper, we consider the influence maximization problem in evolving social networks, and propose an incremental algorithm, IncInf, to efficiently identify top- $K$  influential nodes in dynamic social networks. Taking advantage of the structural evolution of networks and previous information on individual nodes, IncInf substantially reduces the search space and adaptively selects influential nodes in an incremental way. Extensive experiments demonstrate that IncInf significantly reduces the execution time of state-of-the-art static influence maximization algorithm while maintaining satisfying accuracy in terms of influence spread.

There are several future directions for this research. First, IncInf has large potential to fit into modern parallel computing framework. This is because IncInf restricts the computation of influence spread changes into local regions, which could ease the partition of social graph for parallel computation. Moreover, the proposed pruning strategy could be effectively performed in parallel. Second, our current IncInf algorithm is derived from the basic IC model. We believe the conception of incremental computation for influence maximization could be properly extended to other influence diffusion model, such as another classic LT model. Third, although there have been a few research [5, 29] about how to measure the propagation probability, however this problem is not yet well addressed especially for large-scale dynamic social networks.

## References

- [1] Aggarwal, C., Lin, S., Yu, P.: On Influential Node Discovery in Dynamic Social Networks. In: Proceedings of the 12th SIAM International Conference on Data Mining, pp. 636–647. SIAM, California, USA (2012)

[2] ArXiv NetHEPT dataset.  
<http://www.cs.cornell.edu/projects/kddcup/datasets.html>  
(2003) Accessed on 18 June 2013

[3] Barabasi, A.-L., Albert, R.: Emergence of Scaling in Random Networks. *Science* **286**(5439), 509-512 (1999).

[4] Capocci, A., Servedio, V.D.P., Colaiori, F., Burjol, L.S., Donato, D., Leonardi, S., Caldarelli, G.: Preferential attachment in the growth of social networks: the case of Wikipedia. CoRR arXiv preprint arxiv:physics/0602026,  
<http://arxiv.org/abs/physics/0602026/>, (2006)

[5] Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.: Measuring User Influence in Twitter: The Million Follower Fallacy. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media, AAAI, Washington, DC, USA (2010)

[6] Chen, W., Lu, W., Zhang, N.: Time-Critical Influence Maximization in Social Networks with Time-Delayed Diffusion Process. In: Proceedings of the 26th Conference on Artificial Intelligence, Toronto, Canada (2012)

[7] Chen, W., Wang, Y., Yang, S.: Efficient Influence Maximization in Social Networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199-208. ACM, Paris, France (2009)

[8] Chen, W., Wang, C., Wang, Y.: Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029-1038. ACM, Washington, DC, USA (2010)

[9] Chen, Y., Krause, A.: Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization. In: Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA (2013)

[10] Cheng, S., Shen, H., Huang, J., Chen, W., Cheng, X.: IMRank: Influence Maximization via Finding Self-Consistent Ranking. In: Proceedings of the 37th ACM SIGIR Conference, pp. 475-484. ACM, Queensland, Australia (2014)

[11] Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57-66. ACM, San Francisco, CA, USA (2001)

[12] Easley, D., Kleinberg, J.: Power Laws and Rich-Get-Richer Phenomena. In: Networks, Crowds, and Markets: Reasoning about a Highly Connected World. pp. 543-560. Cambridge University Press, (2010)

[13] Gomez-Rodriguez, M., Scholkopf, B.: Influence Maximization in Continuous Time Diffusion Networks. In: Proceedings of the 29th International Conference on Machine Learning. IEEE, Edinburgh, Scotland, UK (2012)

[14] Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: A Data-Based Approach to Social Influence Maximization. In: Proceedings of the 38th International Conference on Very Large Data Bases, pp. 73-84. Istanbul, Turkey (2012)

[15] Goyal, A., Lu, W., Lakshmanan, L.V.S.: CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. In: Proceedings of the 20th International World Wide Web Conference, pp. 47-48. Hyderabad, India (2011)

[16] Habiba, Berger-Wolf, T.Y.: Maximizing the Extent of Spread in a Dynamic Network. DIMACS TR: 2007-20. University od Illinois at Chicago, (2007)

[17] Jiang, Q., Song, G., Cong, G., Wang, Y., Si, W., Xie, K.: Simulated Annealing Based Influence Maximization in Social Networks. In: Proceedings of the 25th AAAI International Conference on Artificial Intelligence, pp. 127-132. AAAI, San Francisco, CA, USA (2011)

[18] Jung, K., Heo, W., Chen, W.: IRIE: A Scalable Influence Maximization Algorithm for Independent Cascade Model and Its Extensions. CoRR arXiv preprint arXiv:1111.4795,  
<http://arxiv.org/abs/1111.4795/>, 1-20 (2011)

[19] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the Spread of Influence through a Social Network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137-146. ACM, Washington, DC, USA (2003)

[20] Kumar, R., Novak, J., Tomkins, A.: Structure and Evolution of Online Social Networks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 611-617. ACM, Philadelphia, Pennsylvania, USA (2006)

[21] Lady Gaga Twitter status,  
<http://twittercounter.com/ladygaga/> (2012), Accessed 15 Dec. 2013

[22] Lee, J., Chung, C.: A Fast Approximation for Influence Maximization in Large Social Networks. In: Proceedings of the International World Wide Web Conference, pp. 1157-1162. IW3C2, Seoul, Korea (2014)

[23] Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic Evolution of Social Networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 462-470. ACM, Las Vegas, Nevada, USA (2008)

[24] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective Outbreak Detection in Networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420-429. ACM, San Jose, CA, USA (2007)

[25] Liu, X., Li, S., Liao, X., Wang, L., Wu, Q.: In-Time Estimation for Influence Maximization in Large-Scale Social Networks. In: Proceedings of the ACM EuroSys Workshop on Social Network Systems, pp. 1-6. ACM, Bern, Switzerland (2012)

[26] Liu, X., Li, M., Li, S., Peng, S., Liao, X., Lu, X.: IMGPU: GPU-Accelerated Influence Maximization in Large-Scale Social Networks. In: IEEE Transactions on Parallel and Distributed Systems, pp. 136-145. IEEE (2014)

[27] Mislove, A., Koppula, H.S., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Growth of the Flickr Social Network. In: Proceedings of the 1st ACM SIGCOMM Workshop on Social Networks, pp. 25-30. Seattle, Washington, USA (2008)

[28] Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61-70. ACM, Edmonton, Canada (2002)

[29] Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Paris, France (2009)

[30] Twitter 2012 Facts and Figures,  
<http://www.website-monitoring.com/blog/2012/11/07/> (2012), Accessed 11 Dec. 2012

[31] Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the Evolution of User Interaction in Facebook. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks, pp. 37-42. Barcelona, Spain (2009)

[32] Wang, Y., Cong, G., Song, G., Xie, K.: Community-based Greedy Algorithm for Mining Top-K Influential Nodes in Mobile Social Networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1039-1048. ACM, Washington, DC, USA (2010)

[33] Zhuang, H., Sun, Y., Tang, J., Zhang, J., Sun, X.: Influence Maximization in Dynamic Social Networks. In : Proceedings of the 13th IEEE International Conference on Data Mining, pp. 1313-1318. IEEE, Dallas, Texas, USA (2013)